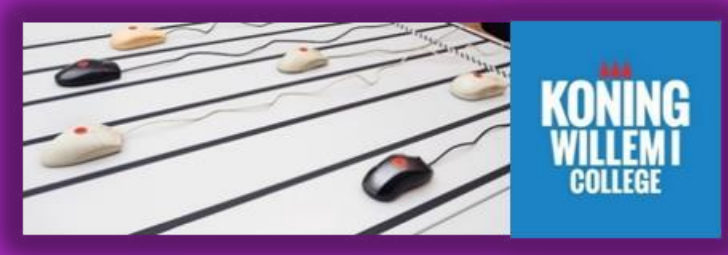
Gevorderd standalone

Testen & Verbeteren

hoofdstuk

2

Acceptatietest



Algemene informatie

Onderwerp	Acceptatietest
Leerdoel(en)	1. De student kan vanuit Use Cases een testplan opstellen 2. De student weet het onderscheid tussen de verschillende testmethodes 3. De student leert over de momenten dat een bepaalde testmethode wordt toegepast 4. De student schrijft een testplan dat zowel goede scenario's als ongewenste scenario's bevat
Vereiste voorkennis	1. De student weet wat Use Cases voor een bedrijfssituatie beschrijft 2. De student heeft voorkennis over de verschillende testmethodieken
Kwalificatiedossier	☐ B1-K1-W1: Plant werkzaamheden en bewaakt de voortgang ☐ B1-K1-W2: Ontwerpt software ☐ B1-K1-W3: Realiseert (onderdelen van) software ☒ B1-K1-W4: Test software ☒ B1-K1-W5: Doet verbetervoorstellen voor de software ☐ B1-K2-W1: Voert overleg ☐ B1-K2-W2: Presenteert het opgeleverde werk ☐ B1-K2-W3: Reflecteert op het werk

Inhoudsopgave

Introductie

Het is natuurlijk belangrijk dat een applicatie wordt getest. Een klant verwacht een applicatie die precies doet wat nodig is en alle gegevens kan verwerken die hij wil verwerken. Ook heeft de klant in zijn dagelijkse routine informatie nodig die hij op een eenvoudige manier uit de applicatie wil halen. De klant is een belangrijke spil voor een applicatie en het testen van de functionaliteiten (testen of alles werkt volgens de wensen en eisen) is een belangrijke fase in het ontwikkelen van een applicatie.

Inhoud

Acceptatietest algemeen

→ In het vorige thema heb je al kennisgemaakt met verschillende vormen van testen. In dit hoofdstuk gaan we het hebben over acceptatietesten. We halen de Use Cases erbij en willen het uiteindelijk ook integreren in een DevOps omgeving.

Een acceptatietest is een functionele test, maar is ook een vorm van black box testing en kan ook worden gezien (in de meeste gevallen) als een user interface test. Verder is een acceptatietest ook meestal een handmatige test en geen geautomatiseerde test zoals je dat bij de Unit-test hebt gezien.

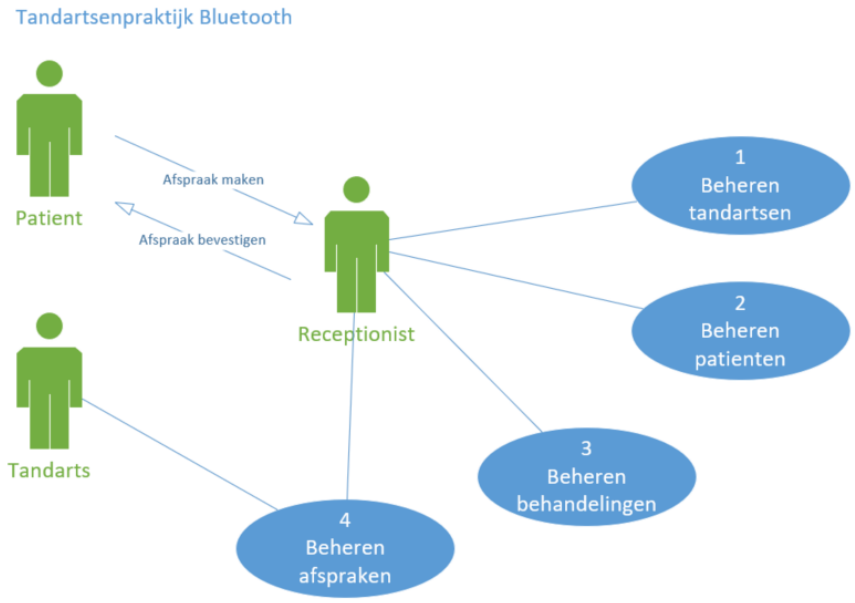
> *Bevestig bovenstaande beweringen door de readers van het vorige thema te herlezen. Weet je nog welke verschillende soorten testen er bestaan? Wat is bijvoorbeeld een regressietest en een localisatietest?*

Een acceptatie test wordt uitgevoerd door een klant of een teamlid of een product owner of iemand die op de hoogte is van welke functionaliteiten er vereist zijn in de applicatie. Een compleet overzicht van functionaliteiten wordt beschreven in de Use Cases die van het systeem zijn opgezet.

Alle processen en sub-processen vormen dus een goede uitgangspunt voor het opstellen van een testplan (=acceptatie testplan).

Use Cases en opstellen van een testplan

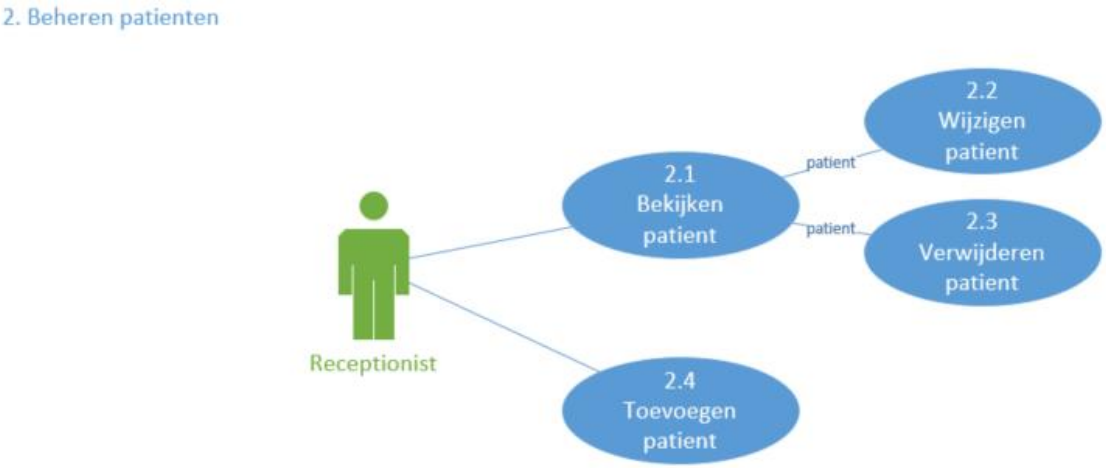
We halen de Use Cases van de tandartsenpraktijk weer tevoorschijn. Voor iedere activiteit in een subproces wordt een apart testcase aangemaakt. Alle testcases samen vormen een testplan.



Hierboven de hoofdprocessen van de tandartsenpraktijk zoals je dat in het vorige thema hebt kunnen zien. Hiervan kunnen we **geen** testplan schrijven omdat de details hiervan ontbreken.

We hebben van ieder hoofdproces de verschillende sub-processen nodig om een testplan te kunnen schrijven.

Als voorbeeld nemen we het Use Case diagram van hoofdproces 2 (Beheren patiënten).



Ieder testscenario moet worden voorzien van concrete waarden die gebruikt worden bij het uitvoeren van de test. Omdat we concrete waarden gaan gebruiken bij het testen is de datadictionary of de create table script ook hierbij belangrijke informatie.

Diagram 1				Patient		
Sleutels			Attribuutnaam	Datatype	Lengte	Voorwaarde invulwaarde
PK	AK	FK				NL / OPT / AUTO(x,x)
PK1			Patientnummer	Numeriek	6	Niet leeg, AUTO(100000,1)
			Aanhef	Karakters	4	Niet leeg
			Voorletters	Karakters	5	Niet leeg
			Achternaam	Karakters	40	Niet leeg
	A1		Emailadres	Karakters	50	Niet leeg
			Geboortedatum	Datum		Niet leeg
	A2		Telefoonnummer	Karakters	10	Optioneel

Toelichting sleutels:

Primary key:	PK1	PK_Patient
Alternate key:	A1	AK_Patient_Emailadres
	A2	AK_Patient_Telefoonnummer
Foreign keys:		

Testing for the good and the bad

Voor ieder testscenario verzin je concrete waarden (input set) waarbij een functionaliteit succesvol wordt afgerond (the good) en verzin je concrete waarden waarbij een functionaliteit NIET succesvol (the bad) wordt afgerond (en waarbij de gebruiker een betekenisvolle terugkoppeling krijgt van de applicatie).

Van ieder van deze sub-processen (4 in aantal) gaan we een apart testcase schrijven. Voor bovenstaande gedeelte zul je dus 4 x 2 (1 good + 1 bad) = 8 testcases moeten opzetten.

Hieronder volgt een voorbeeld voor de testcase van het sub-proces "2.4 Toevoegen patient"

Gebruik voor iedere testcase de volgende tabel als template.

Testcase		
<nr >– <subproces>		
Stap	Actie	**Verwacht resultaat**
1.		
2.		
3.		
4.		
5.		
6.		
Gebruikte parameter waarden:		

Leeg template voor 1 testcase, Het aantal stappen kan per testcase verschillen.

Testcase		
2.41 Toevoegen Patient		
Stap	Actie	**Verwacht resultaat**
1.	Open het beheer patiënten scherm	Een scherm met een overzicht van patiënten wordt getoond.
2.	Klik op de button 'Toevoegen'	Een nieuw scherm wordt geopend met een leeg formulier met patientgegevens wordt getoond
3.	Klik op het eerste Vul alle gegevens in zoals hieronder bij waarden is vermeld	Ieder veld kan worden ingevuld muv het patientennummer
4.	Gebruiker klikt op Opslaan	De melding "Patient is succesvol opgeslagen"
5.	Klik op OK	Het formulier wordt gesloten en de gebruiker keert terug in het overzichtsscherm
Gebruikte parameter waarden:		
Aanhef: Mevrouw Voorletters: V.M. Achternaam: Verbraets Emailadres: v.verbraets@xs4all.nl Geboortedatum: 19-04-1969 Telefoonnummer: 0624567633		

Voorbeelduitwerking testcase Toevoegen Patient succesvol

De tweede testcase voor toevoegen Patient:

Testcase		
2.42 Toevoegen Patient		
Stap	Actie	**Verwacht resultaat**
1.	Open het beheer patiënten scherm	Een scherm met een overzicht van patiënten wordt getoond.
2.	Klik op de button 'Toevoegen'	Een nieuw scherm wordt geopend met een leeg formulier met patiëntgegevens wordt getoond
3.	Klik op het eerste Vul alle gegevens in zoals hieronder bij waarden is vermeld	Ieder veld kan worden ingevuld m.u.v. het patiëntennummer
4.	Gebruiker klikt op Opslaan	De melding "Aanhef en Voorletters zijn verplichte velden"
5.	Klik op OK	De gebruiker keert terug in het formulier en kan dan alsnog de aanhef en voorletters invullen.
Gebruikte parameter waarden:		
Aanhef: leeg Voorletters: leeg Achternaam: Verbraets Emailadres: v.verbraets@xs4all.nl Geboortedatum: 19-04-1969 Telefoonnummer: 0624567633		

Voorbeelduitwerking testcase Toevoegen Patient NIET succesvol

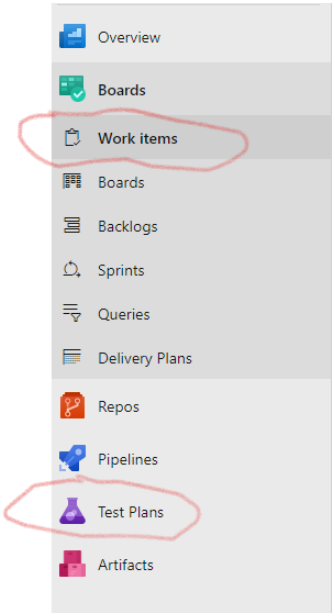
 Je kunt nu **oefening 2.1** maken.

Invoeren van de test in DevOps

De DevOps omgeving biedt uitgebreide mogelijkheden om diverse testen digitaal te beheren, te plannen en uit te voeren.

We beperken ons tot de handmatige testcases die we in het vorige hoofdstuk hebben behandeld en we gaan kijken hoe de testcases in DevOps ingevoerd kunnen worden.

Er zijn 2 plekken waar de testcases ingevoerd kunnen worden. Die kun je in de afbeelding hieronder zien als de rood omcirkelde menuopties.

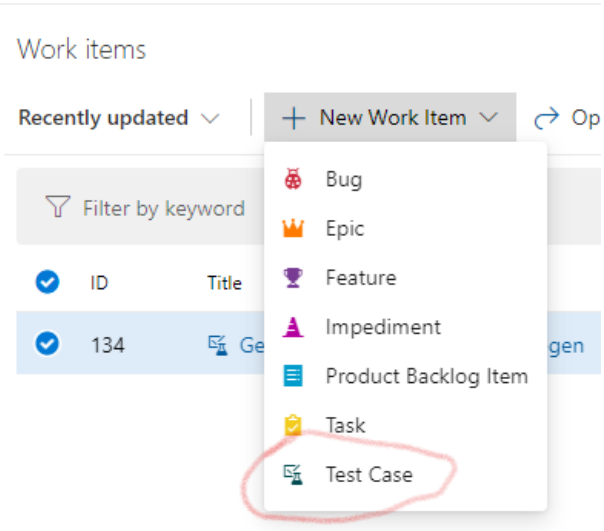


De menuoptie Test Plans is volledig gewijd aan het testen zelf. Een verzameling testcases kan worden gebundeld in een **Test Suite** en als geheel worden gepland, uitgevoerd (laten worden) en testresulaten worden ingevoerd.

Wanneer de test ook echt is uitgevoerd kan iedere testcase **falen** of **succesvol** zoals je dat ook geleerd hebt bij de geautomatiseerde unit test.

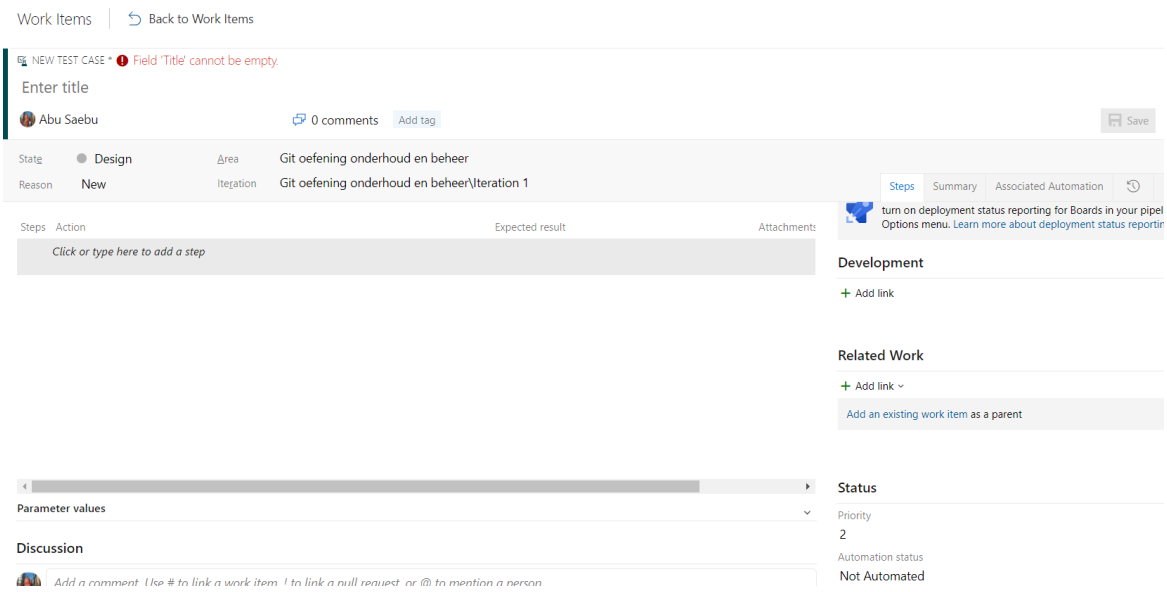
We focussen ons voor nu echter op het invoeren van de testcase in de menuoptie **work-items**.

Voor het in invoeren van een testcase kies je de menuoptie Work items > New Work Item > Test Case.



Alle testgegevens die we in Oefening 2.1 in de tabellen hebben ingevoerd kunnen 1:1 ook in dit formulier worden ingevoerd.

Het lege formulier:



Dit formulier heeft meer invoervelden dan wat we in de vorige paragraaf in de testcase table hebben moeten invoeren. Sommige zijn voor ons niet van toepassing en andere weer wel.

Titel: **TC - <workitem # parent> - De omschrijving van de functionaliteit**

(Het workitem nummer van de parent is in de oefening 2.2 niet bekend, gebruik het subnummer van de usecase)

State: nvt

Reason: nvt

Area: nvt

Iteratie: Vul hier de iteratie/sprint in waarin deze zal worden uitgevoerd (ingepland)

Steps en Actions: Dit kan je overnemen uit de tabel uit de vorige paragraaf

Parameter values: Dit is de set met gegevens die je gaat gebruiken in de test

Discussion: Eventuele bijzonderheden die betrekking hebben op de test

Development: nvt

Related work: Koppel hier de workitem waarin deze functionaliteit is gerealiseerd. (Voor oefening 2.2 is dit niet van toepassing)

Status:

priority status: nvt

Automation status: Not automated (=handmatig)

Parameter values kun je op de volgende manier toevoegen aan je testcase

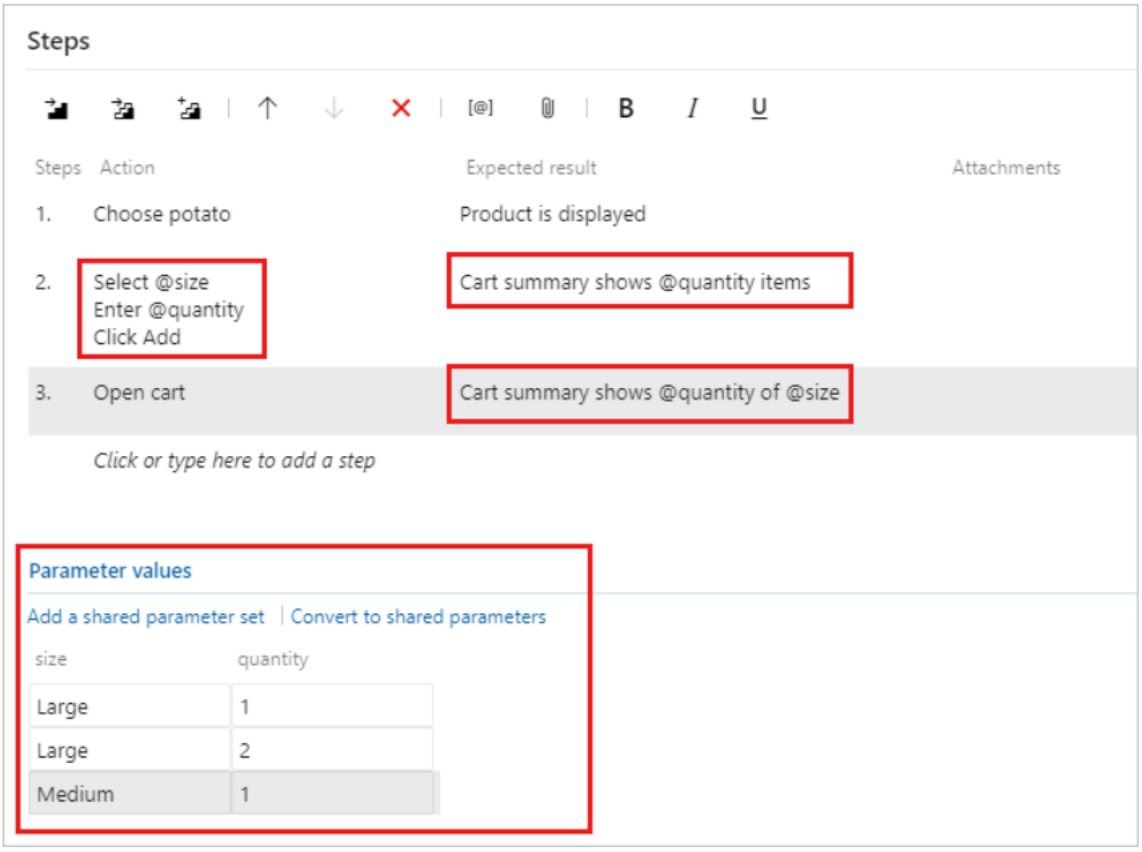


Figure 1, toevoegen parameter values

Op het moment dat in de steps (action/expected result) een verwijzing maakt naar een parameter (@) kun je onder het kopje Parameter values ook concrete waardes toevoegen, waarmee er ook wordt getest bij het uitvoeren van de test (Oefening 2.3)

Tip: Als je de ingevulde parameter set ook 'convert to shared parameter' kan je dezelfde set ook meenemen naar een andere testcase.

Je kunt nu **oefening 2.2** maken.

Uitvoeren van de acceptatietest

De acceptatietest wordt in de praktijk NIET uitgevoerd door de developer zelf maar vaak door een tester of door een eindgebruiker zelf. De tester weet dus niet hoe de applicatie vanuit de code kant eruit ziet maar weet als eindgebruiker precies wat hij/zij nodig heeft van de applicatie. Omdat de tester alleen de buitenkant ziet van de applicatie noem je deze test ook wel een **black box test**.

Het doel van dit soort testen is om de applicatie meer in overeenstemming te laten zijn met de verwachting van een eindgebruiker.

Pakken we nu weer het voorbeeld op die we eerder in deze reader ook gebruikt hebben. De tabel is verder opgedeeld en de verschillende kleuren geven een andere onderdeel aan.

Het groene gedeelte hieronder is de testcase opgesteld door de Developer.

Het blauw gedeelte hieronder wordt ingevuld door de Tester.

Het oranje gedeelte wordt weer ingevuld door de Developer

Testcase		
2.41 Toevoegen Patient		
Stap	Actie	**Verwacht resultaat**
1.	Open het beheer patiënten scherm	Een scherm met een overzicht van patiënten wordt getoond.
2.	Klik op de button 'Toevoegen'	Een nieuw scherm wordt geopend met een leeg formulier met patientgegevens wordt getoond
3.	Klik op het eerste Vul alle gegevens in zoals hieronder bij waarden is vermeld	Ieder veld kan worden ingevuld muv het patientennummer
4.	Gebruiker klikt op Opslaan	De melding "Patient is succesvol opgeslagen"
5.	Klik op OK	Het formulier wordt gesloten en de gebruiker keert terug in het overzichtsscherm
Gebruikte parameter waarden:		

Aanhef: Mevrouw Voorletters: V.M. Achternaam: Verbraets Emailadres: v.verbraets@xs4all.nl Geboortedatum: 19-04-1969 Telefoonnummer: 0624567633		
Tester: K.de Tester	**Testdatum: 23/06/2022**	
Werkelijk resultaat:		
4) Er wordt geen melding getoond dat de patiënt is opgeslagen. 5) In het overzichtsscherm is niet meteen duidelijk of de patiënt wel of niet is opgeslagen.		
Uitslag test: ~~Succes~~ / Faal		
Verbetervoorstel:		**Prioriteit: 2**
Laat een melding verschijnen wanneer de patiënt in de database is opgeslagen. Laat in het overzicht de patiënt meteen zien. De sortering van het overzicht van patiënten moet alfabetisch op achternaam zijn.		
Benodigde aanpassingen	**Developer: DL**	**Tijdsinschatting: 2**
In de Create() method een melding laten tonen nadat de query succesvol is uitgevoerd. Toon ook een melding als er een SQL error optreedt. Het overzicht van patiënten moet worden ververst nadat het patiënten formulier is gesloten.		

Het testen gebeurt in 3 stappen, maar het kan ook voorkomen dat na de aanpassingen en opnieuw testen weer een nieuw verbetervoorstel wordt voorgesteld. Dan ontstaat er een iteratief proces.

De 3 stappen zijn dus:

Opstellen van een test case door een developer

Uitvoeren van de testcase stappen in versie van de applicatie en het beschrijven van het werkelijk resultaat. Als de test faalt wordt het oranje gedeelte ingevuld door een Developer.

Het verbetervoorstel is een niet technische omschrijving van de verbetering en de prioriteit wordt ervan ingeschat.

De benodigde aanpassingen (juist wel een technische omschrijving) worden toegewezen aan een developer en er wordt een tijdsinschatting gemaakt voor de benodigde aanpassingen.

 Je kunt nu **oefening 2.3** maken.